Paralelizando la Búsquedas en los Servidores Web
Línea de investigación: Distribución y Paralelismo

V. Gil Costa, M. Printista
Departamento de Informática
Universidad Nacional de San Luis
San Luis, Argentina
{gvcosta,mprinti}@unsl.edu.ar

M. Marín
Departamento de Computación
Universidad de Magallanes
Punta Arenas, Chile
mmarin@ona.fi.umag.cl

Introducción y Motivación

En la actualidad, la Web se ha convertido en una herramienta fundamental en cualquier tipo de organización, empresa o individuo, los cuales se ven directamente beneficiados con la utilización de este recurso. Debido al continuo aumento en la utilización de Internet y de la migración casi total de los sistemas tradicionales de la arquitectura Web, es que se invierten grandes cantidades de recursos, ya sean humanos y/o económicos, en optimizar la utilización de los recursos existentes, así como también en prestar la mayor cantidad de servicios a los usuario.

Una de las herramientas más importantes para recuperar información desde la Web son las máquinas de búsquedas. La mayoría de estas máquinas utilizan técnicas de recuperación de información para rankear las páginas Web en un cierto orden de relevancia para una determinada consulta. Comparados con los sistemas de recuperación de información bibliográfica de los años 70 y 80, los nuevos motores de búsqueda deben tratar con información más heterogénea, variada en calidad, y mas distribuida y relacionada entre sí.

En los ambientes Web actuales, las consultas suelen ser cortas (1-2 palabras), y la base de datos potencial es muy grande y crece rápidamente. Se estima que la Web posee un rango desde 500 millones hasta un billón de páginas Web, donde muchas de estas páginas son portales a otras bases de datos (la Web escondida).

En respuesta a esta enorme expansión de potenciales fuentes de información, las actuales máquinas o motores de búsqueda Web han enfatizado la velocidad de respuesta a las consultas, dando menos importancia a la efectividad de las respuestas. Debido a esto, varios estudios de nuevas estrategias se han llevado a cabo para satisfacer esta demanda a través del procesamiento paralelo; este procesamiento paralelo ha demostrado ser un paradigma que permite mejorar los tiempos de ejecución de los algoritmos.

Para lograr un procesamiento de consultas eficiente, se deben utilizar técnicas de indexación especializadas sobre grandes colecciones de documentos. Existe un gran número de distintas técnicas de indexación para la recuperación de información en la literatura que han sido implementados bajo distintos escenarios. Algunos ejemplos son los arreglos de sufijos, listas invertidas, y archivos de firmas [1]. Cada uno de ellos tiene sus puntos débiles y fuertes. Sin embargo, debido a su simplicidad y buena performance, los índices o listas invertidas han sido tradicionalmente las técnicas de indexación mas populares utilizadas a través de los años.

Suponiendo una colección de texto compuesta por un gran conjunto de documentos, una lista invertida es básicamente una tabla de vocabulario que mantiene todas las palabras relevantes encontradas en el texto, y una lista asociada por cada una de estas palabras que registra las ocurrencias de las mismas en el texto (identificador del documento y otra información utilizada para rankear las respuestas a los usuarios) [2].

Debido a que los usuarios no entienden exactamente el significado de las búsquedas usando un conjunto de palabras, y pueden obtener respuestas inesperadas, porque no están consientes del punto de vista lógico del texto adoptado por el sistema y finalmente porque tienen dificultades con la lógica booleana, es que los algoritmos de búsqueda mas populares utilizan el modelo de clave única denominado modelo vectorial [1].

1 Trabajos Relacionados

El trabajo propuesto en [15] compara el impacto de performance del procesamiento de consultas, utilizando dos diferentes organizaciones para las listas invertidas. Propone dos opciones básicas para el ordenamiento de los índices: índice de disco e índice de sistema. En la organización de disco, mejor conocida como organización local, los documentos son uniformemente particionados en conjuntos, uno para cada disco, y en cada partición se construyen los índices invertidos para los documentos que residen allí. En la organización de índice de sistema, conocida como organización global, el índice completo es uniformemente distribuido a través de los discos del sistema. El tipo de consulta utilizada es booleana y la arquitectura es de tipo LAN, donde el número de CPUs, el número de controladores de entrada/salida por CPU, y el número de discos varía.

El trabajo realizado en [7] utiliza el modelo probabilístico, un modelo que no toma en cuenta la frecuencia de ocurrencia de cada término del índice en el documento [1]. También se considera que los usuarios tienden a enviar pequeñas cantidades de consultas, lo cual implica que sólo uno o dos procesadores del sistema estarán prestando servicio, y los restantes estarán ociosos.

Los trabajos presentados en [11, 12] utilizan el modelo vectorial para realizar la operación de ranking, y los experimentos son llevados a cabo con la colección TREC-3 [4] y su conjunto de consultas. También implementa y evalúa la performance del procesamiento de consultas sobre casos reales y contrastados. En estos trabajos se propone la estrategia de índices particionados implementados a través de la librería PVM, pero sólo se resuelve una consulta a la vez.

También en los trabajos [2, 3, 5] se presenta un enfoque paralelo, utilizando el modelo BSP, de la organización local y global para las listas invertidas, realizando un estudio de la performance de dichas estrategias de indexación, y un análisis analítico basado en el modelo de costo proporcionado por el modelo de computación paralela *Bulk-Synchronous Parallel, BSP* [16].

2 Nuestra Propuesta

Debido al volumen de los datos en la Web y a que el tráfico de consultas sobre esos datos es por lo general muy grande, incluso en los buscadores especializados a un país determinado, es necesario utilizar técnicas de computación paralela para mejorar la performance de las estrategias de búsqueda.

Por esto, nuestro estudio se centra dentro del marco de los servidores Web, en la operación de búsqueda mediante el uso de índices invertidos o listas invertidas para obtener servidores que sean eficientes y escalables. Para ello, se estudian y analizan nuevas alternativas a las estrategias de indexación existentes, utilizando el paradigma de computación paralela.

Un aspecto clave para ello, es tomar como base la utilización de Modelos Modernos de Computación Paralela [8]. Es decir modelos que sean portables y que entreguen la posibilidad de predecir el desempeño de algoritmos, y que a la vez sean eficientes en la práctica. Uno de estos modelos es el llamado modelo BSP de computación paralela, el cual ha llegado a ser un estándar aceptado ampliamente [16, 14, 17, 18]. BSP reúne la práctica de computación paralela en base a la sincronización global de procesadores, y define una manera estructurada de diseño de algoritmos que facilita la predicción de desempeño y simplifica significativamente el diseño e implementación del software.

Actualmente se están estudiando las estrategias de búsquedas mediante el análisis del diseño de servidores que trabajan sobre bases de datos de textos distribuidas, y sus aplicaciones prácticas. Para ello se toma como punto de partida, las estrategias de indexación local y global realizadas en trabajos anteriores [2, 3, 5], y se buscan nuevas estrategias que permitan acelerar los tiempos de respuestas de los servidores, así como reducir el espacio en memoria principal requerido para mantener dichas estructuras y conocer la unidad de trabajo para poder predecir la carga de trabajo que tendrá cada procesador cuando se realice la distribución de las consultas.

La necesidad de plantear nuevas estrategias, surge porque tanto la organización local como la global presentan falencias. Cuando se trabaja con la indexación local, el problema aparece si la base de datos que se utiliza es pequeña, ya que la operación de *broadcast* que esta organización debe realizar, consume la mayor cantidad de tiempo y recursos (la red). En el caso de la indexación global, el problema se presenta cuando los procesadores poseen listas invertidas de tamaños muy variables, con lo cual esta organización se ve afectada por el desbalance de carga provocado por los diferentes largos de las listas.

Una primera solución a estos problemas se presentan en [6]. En este trabajo se presentan las listas invertidas compuestas o *composite inverted lists*, donde se propone una implementación que combina las dos organizaciones de índices mencionadas. Pero esta implementación de listas compuestas es extremista, debido a que ni la indexación local, ni la global son buenas para ciertos casos.

En el trabajo que se está desarrollando, presenta una propuesta específica de implementación BSP de archivos invertidos para un cluster de PCs, con el fin de mejorar los tiempos de respuestas a los requerimientos de los usuarios, y de mantener una buena performance del sistema. Este trabajo está basado en el concepto de buckets, para lograr obtener por un lado, una mejor distribución de los datos y por otro, un punto intermedio de mejor rendimiento, es decir que los algoritmos implementados sean capaces de evitar los problemas de la estrategia de indexación global, de la local y la compuesta.

Los algoritmos desarrollados presentan una situación intermedia entre la estrategia local y la global mediante la utilización de *buckets*, donde cada lista invertida es dividida en un cierto número de estos *buckets* que son uniformemente distribuidos entre los procesadores. El objetivo es alcanzar un alto grado de paralelismo durante las operaciones de disco, requeridas para recuperar la información de las listas invertidas. También estos algoritmos permiten obtener una reducción significante del costo del *broadcast* para las consultas, y por lo tanto pueden escalar

mas eficientemente.

3 Estrategia de Buckets Iterativos

La estrategia de indexación basada en el concepto de *buckets* para implementar las listas invertidas, propone un punto intermedio entre la indexación local y la global, para obtener un mejor rendimiento. A medida que el tamaño de los buckets crece, el comportamiento de esta estrategia se aproxima al de la global. Contrariamente, a medida que los buckets son mas pequeños, se obtiene un comportamiento similar al de la indexación local.

Esta estrategia denominada Buckets Distribuidos en Diferentes Procesadores Iterativos (BD-DPI) utiliza distintos tipos de distribuciones: (a) uniforme secuencial, donde los buckets de las listas asociadas a cada término son distribuidas uniformemente entre los procesadores del servidor; (b) circular, los buckets se distribuyen en forma circular; (c) hash, los buckets se distribuyen utilizando una función de hash y (d) random, donde los buckets se distribuyen aleatoriamente.

La operación de búsqueda en esta estrategia, consiste en recuperar solo una parte de la lista asociada a cada término que aparece en la consulta, para luego enviársela a una máquina del servidor que realizará el proceso de ranking. Durante este proceso ésta última máquina deberá controlar que los documentos no enviados por los procesadores del servidor, no producen una pérdida de precisión en el resultado final.

Conclusiones

El estudio de los buscadores Web es un tema de relevancia, debido a que es una de las herramientas mas utilizadas hoy en día. Los usuarios esperan recibir buenos resultados en un tiempo razonable y las estructuras de datos mas populares para realizar esta tarea son las listas invertidas. El motivo por el cual se ha trabajado con las listas invertidas como estructuras de datos a través de la cual se realiza la indexación, es porque éstas permiten obtener un procesamiento eficiente de consultas a bases de datos textuales.

Existen dos estrategias clásicas para realizar las búsquedas a través de las listas invertidas: la estrategia de indexación local y global. Pero estas dos estrategias presentan diferentes problemas al momento de trabajar sobre bases de datos suficientemente grandes. Por lo tanto, en este trabajo se intenta buscar otras alternativas y nuevas soluciones a las falencias de estas estrategias. Esta búsqueda se basa en el concepto de buckets para mejorar la distribución de la base de datos, mejorando de esta manera el b alance de carga durante el procesamiento de las consultas y reduciendo los tiempos de espera de los usuarios.

Trabajo Futuro

La Web es cada vez mas un inmenso repositorio de información multimedial, es decir, no sólo texto sino también imágenes, sonidos y video. Ésto representa nuevos desafíos en la investigación de temas tales como el diseño e implementación de máquinas de búsqueda capaces de trabajar con datos multimediales. Por otro lado, el volumen de datos y el tráfico de consultas sobre esos datos es por lo general muy grande, incluso en buscadores especializados a un país determinado, lo cual hace evidente la necesidad de utilizar técnicas de computación paralela. La

plataforma de hardware de uso común en este contexto son los clusters de PCs, lo cual conlleva a investigar el diseño de algoritmos capaces de trabajar sobre datos distribuidos con cualidades de eficiencia y escalabilidad. Temas que en el contexto de sistemas multimediales no han sido investigados en profundidad, especialmente en el caso de máquinas de búsqueda para la Web.

Como trabajo futuro se pretende ampliar las estructuras utilizadas para realizar las búsquedas en la Web. Estas estructuras de datos, como las estructuras de árboles (SAT,dSAT,etc.), permiten realizar búsquedas de textos y otra información multimedial como sonidos, videos, etc.

Referencias

[1] R. Baeza and B. Ribeiro. "Modern Information Retrieval". Addison-Wesley. 1999.

[2] G. V. Gil Costa, "Procesamiento Paralelo de Queries sobre Base de Datos Textuales". Universidad Nacional de San Luis. 2003

[3] Verónica Gil Costa, A. Marcela Printista. "Estrategia de Buckets para Listas Invertidas Paralelas". XII Jornadas Chilenas de computación. Arica, Chile. 8-12 de noviembre del 2004.

[4] Donna Hawking. "Overview of the third text retrieval conference". Proceedings of the third text Retrieval Conference (TREC-3), Maryland, U.S.A.,1994, pág. 1-19,NIST Special Publication 500-207.

[5] M. Marin, C. Bonacic y S. Casas. "Analysis of two indexing structures for text databases", Actas del VIII Congreso Argentino de Ciencias de la Computación (CACIC2002). Buenos Aires, Argentina, Octubre 15 - 19, 2002.

[6] M. Marín, "Parallel text query processing using Composite Inverted Lists".In Second Intenational Conference on Hybrid Intelligent Systems (Web Computing Session).IO Press,2003.

[7] A. MacFarlane and J.A. McCann and S. E. Robertson."Parallel Search using Partitionated Inverted Files". In Proceedings of the 7th International Symposium on String Processing and Information Retrieval,La Coruna, Spain. September 2000. IEEE Computer Society.

[8] McColl W.F. "General purpose parallel computing".A.M. Gibbons and P. Spirakis. Cambridge University Press, 1993.

[9] R. Miller. "A library for Bulk Syncrhonous Parallel programming". Proceedings of the BCS Parallel Processing Specialist Group workshop on General Purpose Parallel Computing, Pp 100-108. December, 1993.

[10] M. Persin, J. Zobel, R. Sacks-Davis. "Filteres Document Retriaval with Frequency-Stores Indexes". Journal of the American Society for Information Science, 1996.

[11] Claudine Santos Badue."Processamento Distribuído de Consultas Usando Arquivos Invertidos Particionados", Universidade Federal de Minas Gerais, Istituto de Ciencias Exatas, Departamento de Ciencia de Computaçao,2001.

[12] Claudine Santos Badue and Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto and Nivio Ziviani."Distributed Query Processing Using Partitioned Inverted Files".SPIRE,2001,10-20.

[13] C. Santos Badue, R. Baeza-Yates, B. Ribeiro-Neto, and N. Ziviani. "Concurrent query processing using distributed inverted files". In the 8th. International Symposium on String Processing and Information Retrieval, pages 10-20, 2001.

[14] D.B. Skillcorn and J. Hill and W.F. McColl. "Questions and Answers about BSP".Oxford University Computing Laboratory,PRG-TR-15-96,1996.

[15] Anthony Tomasic and Hector García-Molina. "Performance of Inverted indices in shared-nothing distributed text document information retrieval systems". In Proceedings of the Second International Conference on Parallel and Sitributed Information Systems. San Diego, California, U.S.A.,1993.

[16] L.G. Valiant. "A Bridging Model for Parallel Computation". Communications of the ACM, Vol. 33, Pp 103-111, 1990.

[17] BSP Worldwilde Standard, http://www.bsp-worldwide.org

[18] BSP PUB Library at Paderborn University, http://www.uni-paderborn.de/bsp